OpenMP를 이용한 대용량 연속어 음성인식기의 병렬 구현

최정욱, 성원용
서울대학교 전기공학부
e-mail : jwchoi@dsp.snu.ac.kr, wysung@snu.ac.kr

Parallel Implementation of a Large Vocabulary Continuous Speech Recognizer Using OpenMP

JungWook Choi and Wonyong Sung
School of Electrical Engineering
Seoul National University

Abstract

We have implemented a parallel continuous speech recognizer exploiting the portable OpenMP library. A dynamic workload distribution method is employed in the emission probability computation for good load balancing. In the Viterbi search, however, we partitioned the search network into independent subtrees in offline to reduce memory synchronization overhead. The test was conducted using Wall Street Journal 20k evaluation and test set. We achieved the speed-up of 330% by utilizing four threads. The recognizer runs about 4.8 times faster than the real-time requirement.

I. 서론

다중 코어 시스템(multi-core system)의 등장과 함께 다중 스레드(multi-theade)를 이용한 고속의 대용량 음성인식기(Large Vocabulary Continuous Speech Recognizer, LVCSR)의 구현이 가능해 지면서, 음성 인식기의 활용범위는 기존의 인간-기계간 실시간 인터페이스에서 최근의 비디오 내 음성 데이터의 검색까지 확장되었다. 다중 코어 시스템을 이용해 고속의 음성 인식을 효과적으로 구현하려면 제한된 주 기억 장치의 대역폭을 고려하여 작업량을 균등화하고 메모리 접근 충돌을 줄이는 병렬화 방법이 필요하다. 이와 관련하여 지금까지 여러 가지 연구들이 있었으나[1-2] 특정 플랫폼의 라이브러리를 이용하여 이식성이 부족하거나[1], 단어나 코어의 수가 증가할 경우에 대한 확장성(scalability)이 부족하였다[2].

이 논문은 지식경제부 출연금으로 ETRI와 시스템반도체산업진흥센터에서 수행한 ITSoC 핵심설계인력양성사업과 교육과학기술부의 재원으로 한국학술진흥재단에서 수행하는 BK21 프로젝트의 지원을 받아 수행된 연구입니다.

본 논문에서는 이식성이 뛰어난 OpenMP 라이브러리를 이용해 다중 쓰레드 버전의 음성 인식기를 구현하였다. 단어나 프로세서 수의 증가에 대해 좋은 확장성을 가지기 위해 미세 병렬 처리 방법(fine grain parallel processing approach)을 이용하였다. 이때 순차적인 연산을 줄이고, 작업량을 균등하게 하며, 일관성 유지를 위한 오버헤드를 최소화 하였다.

II.음성 인식기의 구조

본 논문에서 쓰인 음성 인식기는 입력 음성에 대한 연속어 음성 인식(continuous speech recognition)을 수행하기 위해 문맥 의존적 은닉 마르코프모델(context dependent Hidden Markov Model, HMM)을 이용했다.

음성 인식기는 매 10ms마다 들어오는 30ms 길이의 음성 파형에 대하여 3단계의 작업을 반복한다. 먼저 입력으로 들어온 음성 파형에 대해 MFCC (Mel-Frequency Cepstral Coefficient)를 계산하여 39차원의 특징 벡터를 구한다. 이후 특정 HMM 상태에서 현재 음성 입력이 발생할 확률인 방사확률(emission probability)을 계산하고, 얻어진 방사확률을 바탕으로 비터비 검색(Viterbi search)을 통해 현재 입력까지의 최적의 상태 열을 계산하고 기록한다. 비터비 검색에는 발음 사전(pronunciation lexicon)과 언어

모델(language model)을 결합한 후 잉여 계산을 줄이기 위해 트리 형태로 만들어진 검색 공간 (tree-lexical search space)을 사용하였다. 실제 음성 인식기의 대부분의 연산시간은 방사확률 계산과 비터비 검색에서 소요된다. 따라서 본 논문에서는 방사 확률 계산과, 비터비 검색을 병렬화 하였다.

Ⅲ. OpenMP를 이용한 병렬화

3.1 방사확률 계산 병렬화

$$\log(b(O_t;s)) = \max_m\{C_m - \frac{1}{2}\sum_{k=1}^{K}\frac{(x_k-\mu_{mk})^2}{\sigma_{mk}^2}\} \quad (1)$$

한 HMM 상태에 대한 방사 확률 계산을 위해서는 식(1)에 나타나듯이 여러 개의 가우시안 확률을 계산한 후 최댓값을 얻어 내어야 한다. 이때 HMM 상태 각각에 대한 계산은 서로 독립적이므로 각 프레임마다 계산해야 하는 HMM 상태의 수를 구한 후 다중 쓰레드의 개수만큼 균등하게 분할하여 쓰레드별로 할당하였다. 각 쓰레드 별로 할당된 HMM 상태의 가우시안 확률들은 독립적으로 연산이 수행된다.

3.2 비터비 검색 병렬화

$$\psi_t(s_j;w) = \max_i\{\psi_{t-1}(s_i;w) + \log(a_{ij})\} + \log(b(O_t;s_j,w)) \quad (2)$$

비터비 검색은 식(2)와 같이 검색 공간 안에서 현재 입력 음성 프레임까지의 최적의 상태 열을 찾아내는 일을 한다. 대용량의 메모리 공간을 규칙적으로 읽기만 했던 방사 확률 계산과는 달리, 비터비 검색에서는 대용량의 공간에 대해 읽고 쓰기를 반복한다. 이때 같은 캐시 라인 안의 연속된 메모리 공간에 대해 여러 개의 쓰레드가 동시에 쓰기 동작을 수행할 경우, 많은 프로세서들이 같은 캐시 라인을 변경하게 되어 캐시 일관성(cache coherence) 유지를 위한 동작들을 유발시킨다. 이러한 캐시 라인의 잘못된 공유 (false sharing) 문제는 공유 메모리 다중 프로세서 구조의 성능을 매우 저하시키는 요인이다. 따라서 본 논문에서는 사전에(offline) 검색 공간을 구성하는 트리를 다중 쓰레드의 개수만큼 분할하여, 각 쓰레드에서는 사전에 할당된 검색 공간에서의 연산만 수행하도록 하였다. 이와 같은 선분할 방식은 각 쓰레드에서 접근하는 메모리 공간을 완벽히 분리시킬 수 있기 때문에 캐시 라인을 공유하는 문제를 예방할 수 있다.

Ⅳ. 실험 결과

4.1 실험 환경

본 논문에서는 병렬 수행 처리 성능을 분석하기 위하여, 인텔의 Core i7 프로세서 기반의 시스템을 사용하였다. 대상 시스템은 2.66GHz의 속도를 가진 4개의 코어를 가지고, 코어당 256KB의 L2캐시를 갖추었다. 8MB의 L3 캐쉬는 4개의 코어가 공유한다. 또한 1066MHz로 동작하는 DDR3가 주 메모리로 사용되었다.

음성 인식에 사용된 음향 모델은 화자 독립적인 (speaker independent) 월 스트리트 저널 1 전집을 데이터를 이용하여 오픈소스 음성 인식 툴킷인 HTK로 훈련(trained)한 것이다. 음향 모델은 약 3,000개의 HMM 상태로 구성되어 있으며 각 상태는 16개의 가우시안 분포로 이루어져 있다. 바이그램(bi-gram) 언어 모델을 사용하였으며, 월 스트리트 저널 20,000단어 연속어 음성 인식 태스크(task)로 음성 인식을 테스트 하였다. 인식기의 단어 오 인식률은 약 11.9%이다.

4.2 병렬화에 따른 속도 향상

쓰레드의 개수를 1, 2, 4개로 증가시켜 가면서 병렬 음성 인식기의 성능을 분석하여 표.1에 나타내었다. 쓰레드를 2개를 사용하였을 때 병렬화 되지 않은 음성 인식기에 비하여 약 1.9배의 성능향상을 얻을 수 있었으며 4개의 쓰레드를 사용하였을 경우에는 약 3.3배의 성능 향상을 얻을 수 있었다. 4개의 쓰레드를 사용한 병렬 처리 음성인식기의 경우 실시간 요구량 보다 약 4.8배 따른 속도로 동작하는 것을 확인 할 수 있다.

	방사확률	비터비 검색	합계	속도 향상
1 쓰레드	1,033	830	1,863	**1.00**
2 쓰레드	524	452	976	**1.91**
4 쓰레드	278	281	559	**3.33**

표.1. 다중 쓰레드 개수에 따른 성능 향상
(성능은 1초의 음성은 처리하는데 필요한 MCycles)

Ⅴ. 결론

본 논문에서는 OpenMP를 이용하여 대용량 연속어 음성 인식기를 병렬 구현하였다. 이를 통해 2개의 쓰레드를 이용하였을 때 1.9배, 그리고 4개의 쓰레드를 이용하였을 때 3.3배의 성능 향상을 얻어내었다. 구현된 병렬 음성 인식기는 실시간 요구량보다 약 4.8배 빠른 성능을 보여주었으며, OpenMP를 사용하였기 때문에, 다른 시스템으로의 이전이 용이하다.

참고문헌

[1] S. Phillips and A. Rogers, “Parallel speech recognition,” *Int. Journal of Parallel Programming*, vol. 27, no. 4, pp. 257.288, 1999.

[2] S. Ishikawa, K. Yamabana, R. Isotani, and A. Okumura, “Parallel LVCSR algorithm for cellphone-oriented multicore processors,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 177.180, May 2006.